Sep 23 05 12:04p   ANNE V. DOUGHERTY   9149621973   p.14

Simulated annealing - encyclopedia article about Simulated annealing.   Page 1 of 5

THEFREEDICTIONARY BY FARLEX

Word (phrase): Simulated annealing   Word   L

. Dictionary/ ● Computing   Medical   ● Legal   . Financial ● Acronyms . Columbia   ■ Wikipedia
  thesaurus   dictionary   dictionary   dictionary   dictionary   encyclopedia   encyclopedia

Simulated annealing   Also found in: Computing, Acronyms   0.03 sec.

Simulated annealing (SA) is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space. It was independently invented by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi in 1983, and by V. Černy in 1985.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

Overview

In the simulated annealing (SA) method, each point s of the search space is compared to a *state* of some physical system, and the function $E(s)$ to be minimized is interpreted as the internal *energy* of the system in that state. Therefore the goal is to bring the system, from an arbitrary *initial state*, to a state with the minimum possible energy.

The basic iteration

At each step, the SA heuristic considers some *neighbours* of the current state s, and probabilistically decides between moving the system to state s' or staying put in state s. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically this step is repeated until the system reaches a state which is good enough for the application, or until a given computation budget has been exhausted.

The neighbours of a state

The neighbours of each state are specified by the user, usually in an application-specific way. For example, in the traveling salesman problem, each state is typically defined as a particular *tour* (a permutation of the cities to be visited); then one could define two tours to be neighbours if and only if one can be converted to the other by interchanging a pair of adjacent cities.

Transition probabilities

The probability of making the transition to the new state s' is a function $P(\delta E, T)$ of the energy difference $\delta E = E(s') - E(s)$ between the two states, and of a global time-varying parameter T called the *temperature*.

One essential feature of the SA method is that the transition probability P is defined to be nonzero when δE is positive, meaning that the system may move to the new state even when it is *worse* (has a higher energy) than the current one. It is this feature that prevents the method from becoming stuck in a *local minimum* — a state whose energy is far from being minimum, but is still less than that of any neighbour.

Also, when the temperature tends to zero and δE is positive, the probability $P(\delta E, T)$ tends to zero. Therefore, for sufficiently small values T, the system will increasingly favor moves that go "downhill" (to lower energy values), and avoid those that go "uphill". In particular, when T is 0, the procedure reduces to the greedy algorithm — which makes the move if and only if it goes downhill.

Also, an important property of the P function is that the probability of accepting a move decreases when (positive) δE grows bigger. For any two moves that both have positive δE values the P function favours the smaller value (smaller loss).

When δE is negative, $P(\delta E, T) = 1$. However, some implementations of the algorithm do not guarantee this property with the P function, but rather they explicitly check whether δE is negative, in which case the move is accepted.

Obviously, the effect of the state energies on the system's evolution depend crucially on the temperature. Roughly speaking, the evolution is sensitive only to coarser energy variations when T is large, and to finer variations when T is small.

The annealing schedule

Another essential feature of the SA method is that the temperature is gradually reduced as the simulation proceeds. Initially, T is set to a high value (or infinity), and it is decreased at each step according to some *annealing schedule* — which may be specified by the user, but must end with $T=0$ towards the end of the allotted time budget. In this way, the system is expected to wander initially towards a broad region of the search space containing good solutions, ignoring small features of the energy function; then drift towards low-energy regions that become narrower and narrower; and finally move downhill according to the steepest descent heuristic.

Convergence to optimum

It can be shown that, for any given finite problem, the probability that the simulated annealing algorithm terminates with the global optimal solution approaches 1 as the annealing schedule is extended. This theoretical result is, however, not particularly helpful, since the annealing time required to ensure a significant probability of success will usually exceed the time required for a complete search of the solution space.

Pseudo-code

The following pseudo-code implements the simulated annealing heuristic, as described above, starting from state $s0$ and continuing to a maximum of $kmax$ steps or until a state with energy $emax$ or less is found. The call neighbour(s) should generate a randomly chosen neighbour of a given state S; the call random() should return a random value in the range [0, 1). The annealing schedule is defined by the call temp(r), which should yield the temperature to use, given the fraction r of the time budget that has been expended so far.

```
s := s0
```

Sep 23 05 12:05p   ANNE V.DOUGHERTY   9149621973   p.16

Simulated annealing - encyclopedia article about Simulated annealing.   Page 3 of 5

```
e := E(s)
k := 0
while k < kmax and e > emax
sn := neighbour(s)
en := E(sn)
if en < e or random() < P(en - e, temp(k/kmax)) then
s := sn; e := en
k := k + 1
return s
```

Selecting the parameters

In order to apply the SA method to a specific problem, one must specify the state space, the neighbour selection method (which enumerates the candidates for the next state s'), the probability transition function, and the annealing schedule. These choices can have a significant impact on the method's effectiveness. Unfortunately, there are no choices that will be good for all problems, and there is no general way to find the best choices for a given problem. It has been observed that applying the SA method is more of an art than a science.

State neighbours

The neighbour selection method is particularly critical. The method may be modeled as a *search graph* — where the states are vertices, and there is an edge from each state to each of its neighbours. Roughly speaking, it must be possible to go from the initial state to a "good enough" state by a relatively short path on this graph, and such a path must be as likely as possible to be followed by the SA iteration.

In practice, one tries to achieve this criterion by using a search graph where the neighbours of s are expected to have about the same energy as s. Thus, in the traveling salesman problem above, generating the neighbour by swapping two adjacent cities is expected to be more effective than swapping two arbitrary cities. It is true that reaching the goal can always be done with only $n-1$ general swaps, while it may take as many as $n(n-1)/2$ adjacent swaps. However, if one were to apply a random general swap to a fairly good solution, one would almost certainly get a large energy increase; whereas swapping two adjacent cities is likely to have a smaller effect on the energy.

Transition probabilities

The transition probability function P is not as critical as the neighbourhood graph, provided that it follows the general requirements of the SA method stated before. Since the probabilities depend on the temperature T, in practice the same probability function is used for all problems, and the annealing schedule is adjusted accordingly.

The "classical" formula

In the original formulation of the method by Kirkpatric *et. al*, the transition probability $P(\delta E, T)$ was defined as 1 if $\delta E < 0$ (i.e., downhill moves were always performed); otherwise, the probability would be $e^{-\delta E/T}$. This formula comes from the Metropolis-Hastings algorithm, used here to generate samples from the Maxwell-Boltzmann distribution governing the distribution of energies of molecules in a gas. Other transition rules can be used,

also.

Annealing schedule

The annealing schedule is less critical than the neighbourhood function, but still must be chosen with care. The initial temperature must be large enough to make the uphill and downhill transition probabilities about the same. To do that, one must have some estimate of the value of δE for a random state and its neighbours.

The temperature must then decrease so that it is zero, or nearly zero, at the end of the alloted time. A popular choice is the exponential schedule, where the temperature decreases by a fixed factor $\alpha < 1$ at each step.

See also

- Markov chain
- Combinatorial optimization
- Genetic algorithm
- Ant colony optimization
- Automatic label placement
- Multidisciplinary optimization

References

- S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, Optimization by Simulated Annealing, Science, Vol 220, Number 4598, pages 671-680, 1983. http://citeseer.ist.psu.edu/kirkpatrick83optimization.html.
- V. Cerny, A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. Journal of Optimization Theory and Applications, 45:41-51, 1985

▽ **Mentioned in**   ▽ **References in classic literature**

Annealing	No-Free-Lunch	No references found
Ant colony optimization	theorems	
Automatic label	SAXS	
placement	Simulated annealing	
Boltzmann machine	algorithms	
Combinatorial	Simulated	
optimization	annealing/Temp	
Global optimization	Simulated annealling	
Linesearch	Stochastic tunneling	
Markov chain	Swarm intelligence	
geostatistics	Timeline of algorithms	
Metaheuristic	Trial and error	
Multidisciplinary design	**More results ▶▶**	
optimization		

▽ **Wikipedia browser** ▽ **Full browser**

̂
̄

SimTown   ● Simulated Brillouin
Simtuna Hundred   Scattering
SimTunes   🖫 simulated   ● Simulated

Simula
Simulacra and
Simulation
Simulacron-3
Simulacrum
▶ **Simulated annealing**
Simulated
annealing/Temp
Simulated Emergency
Test
Simulated mechanical
puzzle
Simulated reality
Simulation
Simulation cockpit
Simulation game

⩓

☩
simulacrum
● simulacrum
■ simulacrum
⋄ simular
⋄ simulate
⋄ simulate
simulate
● simulate
● simulate
◙ simulate
■ simulate
· simulated
: simulated
simulated
simulated
● simulated

■ simulated
● Simulated (mission)
Firing
● Simulated Ab-initio
Molecular Orbital
● Simulated Acidic Rain
● Simulated Air Defense
System
● Simulated Aircraft
Maintenance Trainer
● Simulated Airdrop
Training Bundle
▶ **Simulated annealing**
● Simulated annealing
algorithms
■ Simulated annealing
algorithms
● Simulated Annealing
Heuristic
■ Simulated
annealing/Temp
● Simulated annealling
■ Simulated annealling
● Simulated Area
Weapons Effect-Radio
Frequency
● Simulated Area
Weapons Effects

⩔

Comprehensive Air
Navigation
■ Simulated
consciousness
● Simulated Core
● Simulated
Countercurrent Moving-
Bed Chromatographic
Reactor
● Simulated Distressed
Submarine
● Simulated During
Exercise
● Simulated Electronics
Launch Missile
■ Simulated Emergency
Test
● Simulated End-Of-Life
● Simulated
Environment Test Signal
● Simulated
Environmental Test
Signal
● Simulated Fire
● Simulated Flame-Out
● Simulated Flight Path
● Simulated Gastric
Fluid (chemical
dissolution analysis)

⩔

THE FREE **DICTIONARY** **Word (phrase):** Simulated annealing   **Word**   **Look it up**

Free Tools: For surfers: Browser extension | Word of the Day NEW! | Add the dictionary to favorites | Help
For webmasters: Free content NEW! | Linking | Lookup box | Double-click lookup | Partner with us   NO popups spyware

▽ Disclaimer | Privacy policy | Feedback | Copyright © 2005 Farlex, Inc.

GETFIREFOX